where the multicasted streaming interactive video output of the app/game servers **1521-1525** is fed back into the app/game servers **1521-1525** either in real-time via path **1552** or after a selectable delay via path **1551**. This enables a wide range of practical applications (e.g., such as those illustrated in FIGS. **16**, **17** and **20**) that would be either impossible or infeasible through prior art server or local computing architectures. But, as a more general architectural feature, what feedback loop **1550** provides is recursion at the streaming interactive video level, since video can be looped back indefinitely as the application requires it. This enables a wide range of application possibilities never available before.

Another key architectural feature is that the video streams are unidirectional UDP streams. This enables effectively an arbitrary degree of multicasting of streaming interactive video (in contrast, two-way streams, such as TCP/IP streams, would create increasingly more traffic logjams on the networks from the back-and-forth communications as the number of users increased). Multicasting is an important capability within the server center because it allows the system to be responsive to the growing needs of Internet users (and indeed of the world's population) to communicate on a one-to-many, or even a many-to-many basis. Again, the examples discussed herein, such as FIG. **16** which illustrates the use of both streaming interactive video recursion and multicasting are just the tip of a very large iceberg of possibilities.

In one embodiment, the various functional modules illustrated herein and the associated steps may be performed by specific hardware components that contain hardwired logic for performing the steps, such as an application-specific integrated circuit ("ASIC") or by any combination of programmed computer components and custom hardware components.

In one embodiment, the modules may be implemented on a programmable digital signal processor ("DSP") such as a Texas Instruments' TMS320x architecture (e.g., a TMS320C6000, TMS320C5000, . . . etc). Various different DSPs may be used while still complying with these underlying principles.

Embodiments may include various steps as set forth above. The steps may be embodied in machine-executable instructions which cause a general-purpose or special-purpose processor to perform certain steps, Various elements which are not relevant to these underlying principles such as computer memory, hard drive, input devices, have been left out of the figures to avoid obscuring the pertinent aspects.

Elements of he disclosed subject matter may also be provided as a machine-readable medium for storing the machine-executable instructions. The machine-readable medium may include, but is not limited to, flash memory, optical disks, CD-ROMs, DVD ROMs, RAMs, EPROMs, EEPROMs, magnetic or optical cards, propagation media or other type of machine-readable media suitable for storing electronic instructions. For example, the present invention may be downloaded as a computer program which may be transferred from a remote computer (e.g., a server) to a requesting computer (e.g., a client) by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

It should also be understood that elements of the disclosed subject matter may also be provided as a computer program product which may include a machine-readable medium having stored thereon instructions which may be used to program a computer (e.g., a processor or other electronic device) to perform a sequence of operations. Alternatively, the operations may be performed by a combination of hardware and software. The machine-readable medium may include, but is not limited to, floppy diskettes, optical disks, CD-ROMs, and magneto-optical disks, ROMs. RAMs, EPROMs, EEPROMs, magnet or optical cards, propagation media or other type of media/machine-readable medium suitable for storing electronic instructions. For example, elements of the disclosed subject matter may be downloaded as a computer program product, wherein the program may be transferred from a remote computer or electronic device to a requesting process by way of data signals embodied in a carrier wave or other propagation medium via a communication link (e.g., a modem or network connection).

Additionally, although the disclosed subject matter has been described in conjunction with specific embodiments, numerous modifications and alterations are well within the scope of the present disclosure. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.

What is claimed is:

1. A computer-implemented method comprising:

running a real-time application on one or more servers of a hosting service center, wherein the real-time application is a game and wherein the real-time application is being played by a first user of a first client device remote to the hosting service center via compressed streaming interactive video transmitted over the Internet;

continually storing, in a delay buffer of the one or more servers, application state information of the real-time application along with the compressed streaming interactive video as the real-time application is being played, the application state information as stored in the delay buffer is usable to generate additional views of the real-time application for replay; and

running a user interface (UI) application on another server of the hosting service center, the UI application allowing a second user of a second client device remote to the hosting service center to,

rewind based on the application state information;

replay a segment of the played real-time application based on the application state information stored in the delay buffer; and

control a camera view for the replay of the segment of the played real-time application, the control of the camera view causing dynamic generating of video frames by processing the application state information as stored in the delay buffer for one or more additional fly-through perspectives for the segment of the played real-time application;

wherein the one or more additional fly-through perspectives for the segment that is replayed to the second user of the second client device provided by the dynamic generating of video frames are different from a view of the real-time application displayed to the first user on the first client device.

2. The computer-implemented method of claim **1**, wherein the application state information supports a plurality of replays with changes in camera views from different fly-through perspectives.

3. The computer-implemented method of claim **1**, wherein the control of the camera view is accomplished through Digital Video Recorder (DVR) control functionality.